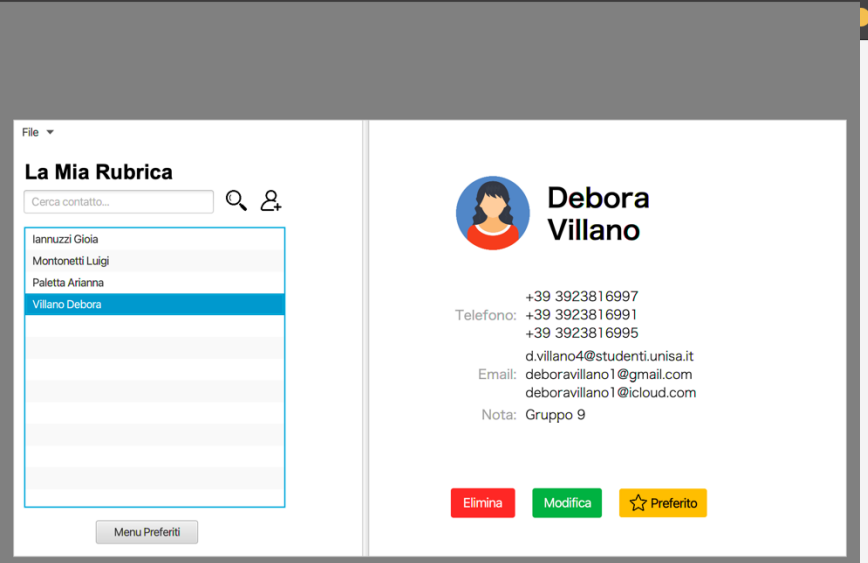
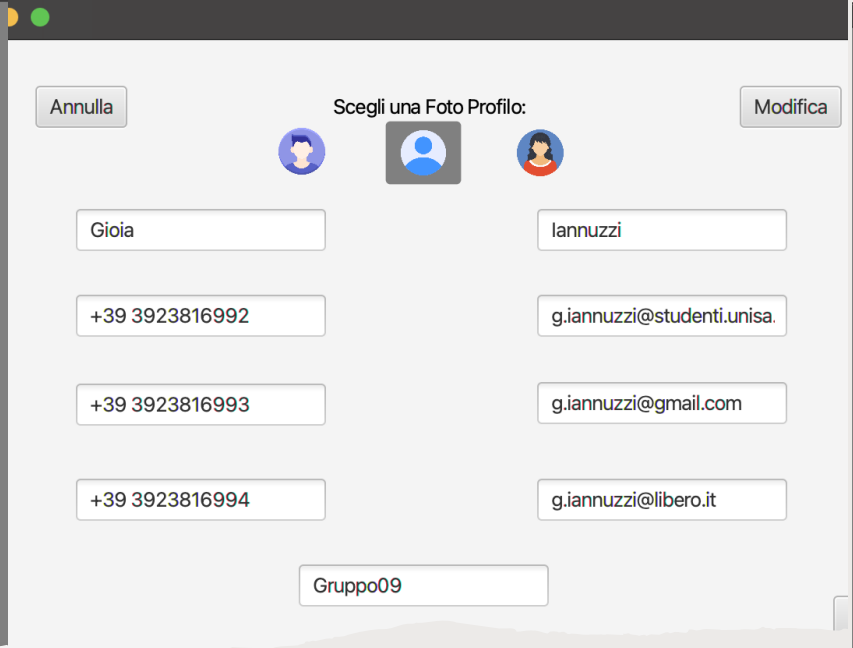
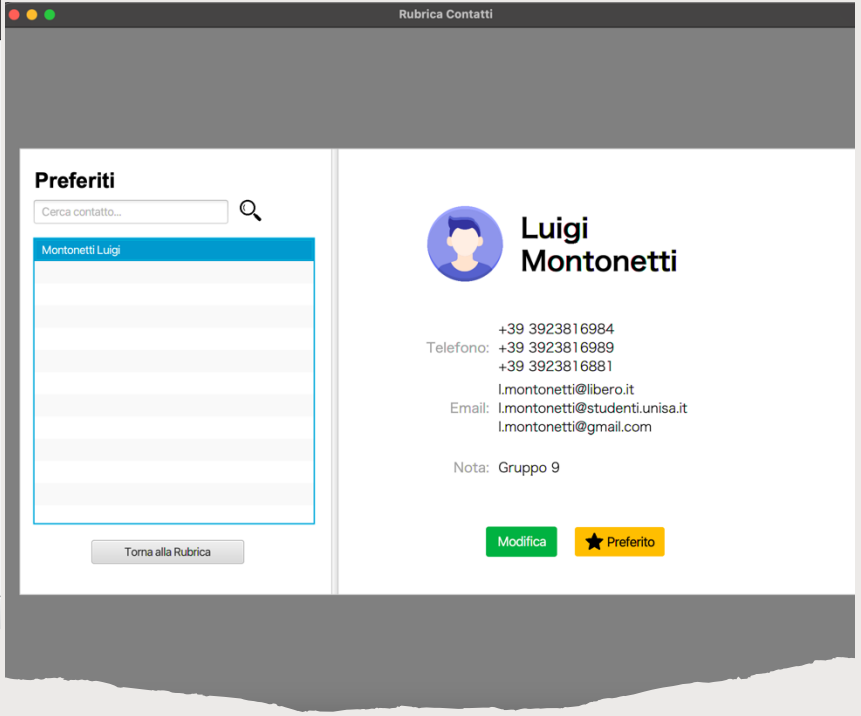






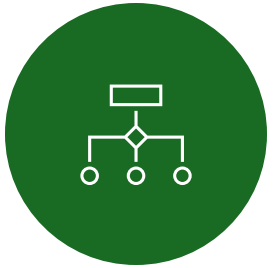
Applicazione Software Rubrica Telefonica

Gruppo 9

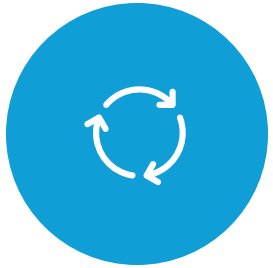
- *0612707416 - Gioia Iannuzzi*
- *0612708273 - Luigi Montonetti*
- *0612707862 - Arianna Paletta*
- *0612708294 - Debora Villano*

Contenuto della discussione

REQUISITI E CASI D’USO

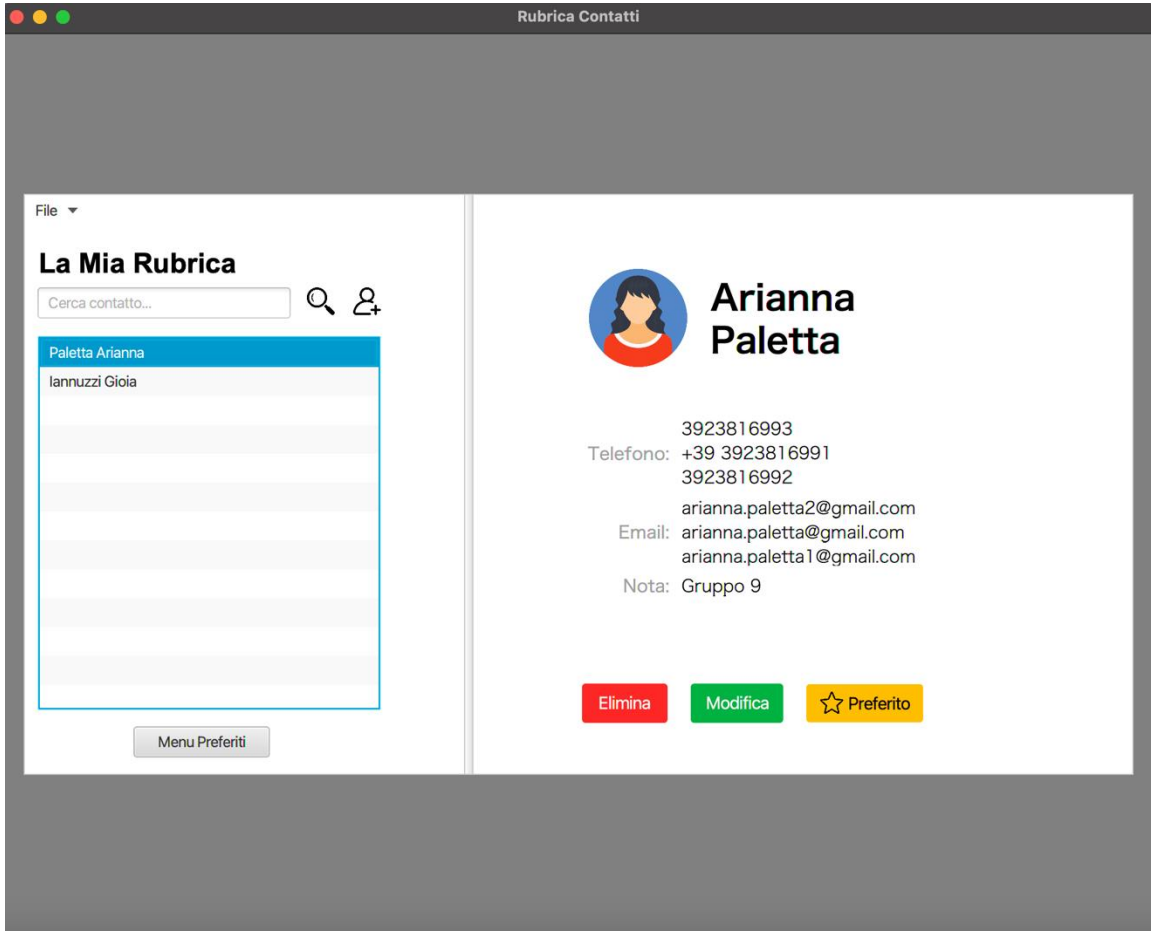


DIAGRAMMI



CAMBIAMENTI IN CORSO D'OPERA

DEMO DEL PROGRAMMA FUNZIONANTE



INTRODUZIONE

- **Proposito**: Definire i la progettazione del sistema *Rubrica Telefonica*.
- **Scopo**: Applicazione per gestire contatti telefonici, compatibile con Windows, macOS e Linux, con funzionalità di ricerca, modifica e organizzazione.
- **Destinatari**: Sviluppatori, tester, utenti finali e amministratori di sistema.

REQUISITI FUNZIONALI

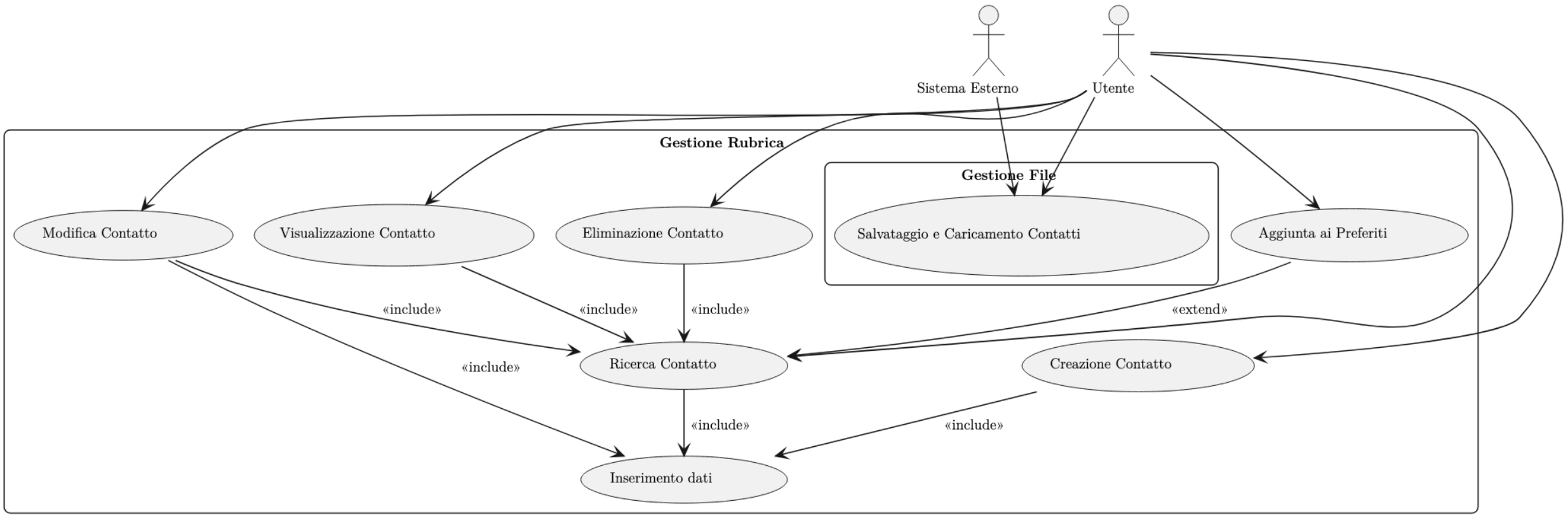
ID	Requisito	Priorità	Breve descrizione	Rischio tecnico
IF-1.1	Creare Contatti	Alta	Deve essere possibile aggiungere nuovi contatti alla rubrica.	Basso
IF-1.2	Modificare Contatti	Alta	È possibile modificare i dettagli di un contatto esistente.	Basso
IF-1.3	Cancellare Contatti	Alta	È possibile rimuovere contatti dalla rubrica chiedendo una conferma.	Basso
IS-1	Salvare e Caricare Contatti	Alta	Le informazioni della rubrica devono essere salvate su file e caricate da file, mantenendo l'intera rubrica.	Medio
BF-1	Validazione dei campi obbligatori	Alta	È obbligatorio inserire un nome e/o un cognome.	Medio
DF-1	Nome e Cognome	Alta	Ogni contatto deve avere un nome e un cognome, ma almeno uno dei due deve essere presente.	Basso
DF-2	Numeri di Telefono	Alta	Ogni contatto può avere da 0 a 3 numeri di telefono.	Basso
IF-2.1	Ricerca di Contatto	Media	È possibile cercare un contatto inserendo la sottostringa iniziale di nome, cognome, numero o mail.	Medio
UI-1	Aggiunta Note di un Contatto	Bassa	È possibile aggiungere delle note ad un contatto.	Medio
IF-2.2	Visualizzazione Contatti	Alta	È possibile visualizzare una lista dei contatti.	Medio
BF-2	Gestione Duplicati	Media	Il sistema richiede conferma per contatti con lo stesso nome, permettendo di sovrascrivere o creare un nuovo contatto con dettagli aggiuntivi.	Medio
UI-2	Aggiunta Foto Contatto	Bassa	È possibile aggiungere una foto ad un contatto.	Alto
UI-3	Interfaccia Grafica	Alta	Il sistema deve disporre di un'interfaccia grafica.	Basso
UI-4	Tastierino di Aggiunta Contatto	Bassa	È possibile salvare un contatto digitando il numero di telefono da tastierino.	Alto
IF-2.3	Menu Preferiti	Media	L'utente può aggiungere contatti alla lista dei preferiti e accedervi rapidamente.	Medio

REQUISITI NON FUNZIONALI

Requisito	Priorità	Categoria	Breve descrizione	Rischio tecnico
L'interfaccia GUI	Alta	Usabilità	L'interfaccia utente deve essere visiva e basata su elementi grafici, facilitando l'interazione.	Basso
Formato numeri	Bassa	Affidabilità	I numeri di telefono devono rispettare un formato internazionale standard (es. +39 123 456 789).	Medio
Formato email	Bassa	Affidabilità	Gli indirizzi email devono essere validi secondo le specifiche RFC 5322.	Medio
Sicurezza dei Dati	Alta	Sicurezza	I dati della rubrica devono essere protetti contro accessi non autorizzati utilizzando un sistema di password o crittografia.	Basso
Limite Preferiti	Bassa	Usabilità	Il sistema deve permettere un massimo di 20 contatti nella lista dei preferiti.	Medio
Supporto Multilingua	Bassa	Accessibilità	L'interfaccia deve essere disponibile in almeno due lingue (es. italiano e inglese) e facilmente estensibile ad altre lingue.	Alto
Comportamento di Ricerca	Alta	Usabilità	La ricerca deve essere case-insensitive e supportare caratteri speciali.	Basso
Backup Automatico	Bassa	Affidabilità	Deve essere possibile configurare un backup automatico giornaliero o settimanale dei dati della rubrica.	Medio
Prestazioni Applicative	Alta	Prestazioni	L'applicazione deve rispondere alle operazioni comuni (es. ricerca, aggiunta, modifica) entro 1 secondo per il 95% dei casi.	Medio

I requisiti sono stati redatti secondo lo standard IEEE Std 830-1993 per guidare progettazione, sviluppo e validazione.

CASI D'USO


Sistema Esterno
Utente
Gestione Rubrica
Gestione File
Modifica Contatto
Visualizzazione Contatto
Eliminazione Contatto
Salvataggio e Caricamento Contatti
Aggiunta ai Preferiti
«include»
«include»
«include»
«extend»
«include»
Ricerca Contatto
Creazione Contatto
«include»
«include»
Inserimento dati

STATO DEI REQUISITI

ID	Priorità	Stato del requisito	Rischio tecnico
IF-1.1	Alta	Accettato per questa release	Basso
IF-1.2	Alta	Accettato per questa release	Basso
IF-1.3	Alta	Accettato per questa release	Basso
IS-1.1	Alta	Accettato per questa release	Medio
IS-1.2	Alta	Rimandato a release successiva	Alto
BF-1	Alta	Accettato per questa release	Medio
BF-2	Media	Accettato per questa release	Medio
DF-1	Alta	Accettato per questa release	Basso
DF-2	Alta	Accettato per questa release	Basso
IF-2.1	Media	Accettato per questa release	Medio
IF-2.2	Alta	Accettato per questa release	Medio
UI-1	Bassa	Accettato per questa release	Medio
UI-2	Bassa	Accettato per questa release	Alto
UI-3	Alta	Accettato per questa release	Basso
UI-4	Bassa	Accettato per questa release	Alto
IF-2.3	Media	Accettato per questa release	Medio

Tabella 10: Tabella di Stato dei Requisiti Funzionali

DESIGN ARCHITETTURALE

- L'architettura del sistema segue il pattern **MVC (Model-View-Controller)** :
  - **Model**: Gestisce i dati e la logica di business.
  - **View**: Rappresenta l'interfaccia utente.
  - **Controller**: Gestisce gli input dell'utente e aggiorna il modello.

View (FXML)	Descrizione	Controller Associato
`InterfacciaUtente.fxml`	Gestione rubrica principale	`InterfacciaUtenteController.java`.
`MenuPreferiti.fxml`	Gestione dei contatti preferiti	`MenuPreferitiController.java`
`InterfacciaAggiungiModifica.fxml`	Aggiunta e modifica contatti	`InterfacciaAggiungiModificaController.java`

Tabella 12: Elenco delle View del Sistema

Classe Model	Descrizione
`Contatto`	Rappresenta un singolo contatto con attributi come nome, cognome, telefono, email e note.
`Rubrica`	Gestisce la lista principale dei contatti e la lista dei preferiti.
`SalvaCaricaRubrica`	Si occupa della persistenza dei dati della rubrica su file JSON o CSV.
`SalvaCaricaPreferiti`	Si occupa della persistenza dei contatti preferiti su file JSON.
`ContattoValidator`	Valida i campi inseriti (nome, email, telefono) e rileva eventuali duplicati.

Tabella 14: Elenco del Model del Sistema

Controller	Descrizione	View Associata
`InterfacciaUtenteController.java`	Gestisce l'interfaccia principale della rubrica, inclusa la visualizzazione dati.	`InterfacciaUtente.fxml`.
`MenuPreferitiController.java`	Gestisce la lista dei contatti preferiti, inclusi aggiornamenti e rimozioni.	`MenuPreferiti.fxml`
`InterfacciaAggiungiModificaControl`	Aggiunta e modifica contatti	`InterfacciaAggiungiModificaController.fxml`

Tabella 13: Elenco dei Controller del Sistema

DIAGRAMMI DI SEQUENZA

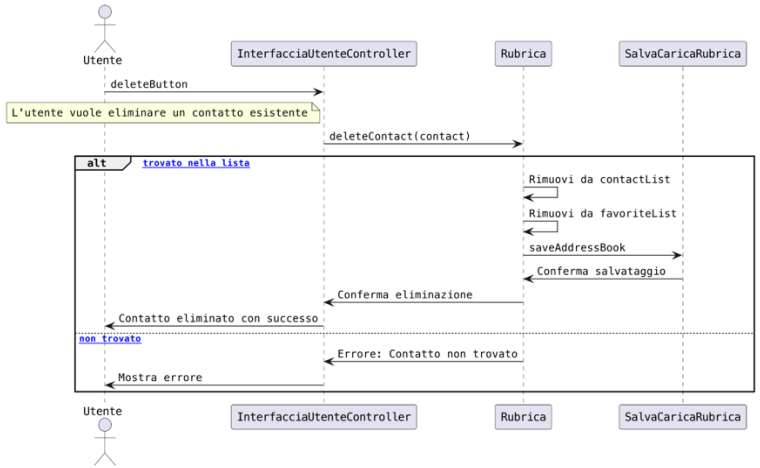


Figura 4: Eliminazione di un Contatto

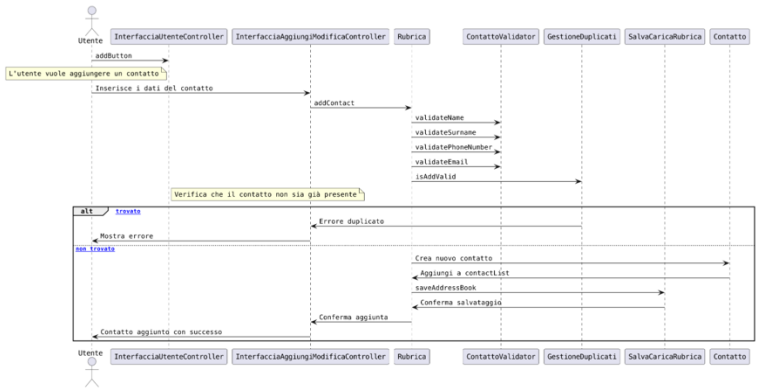


Figura 2: Creazione di un Contatto

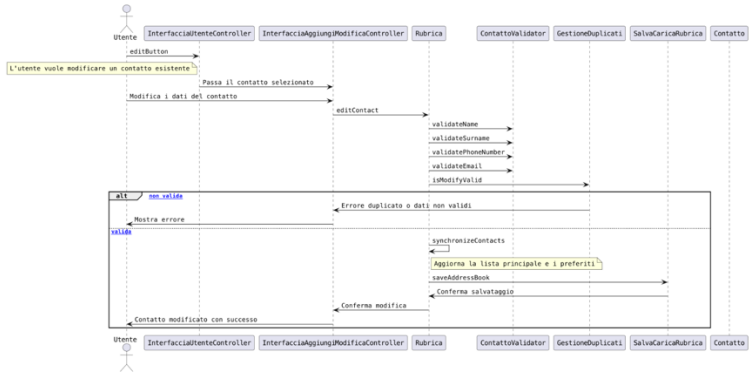


Figura 3: Modifica di un Contatto

DIAGRAMMI DI SEQUENZA

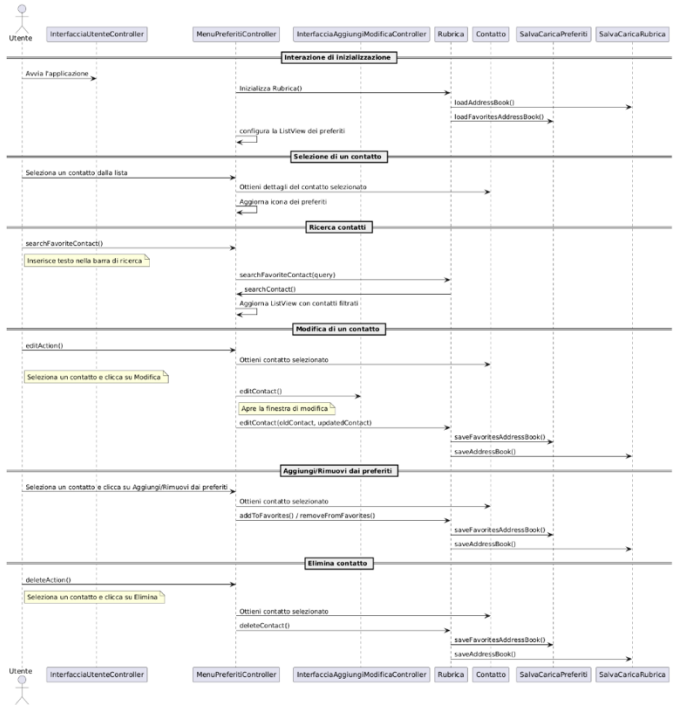


Figura 5: Gestione del Menù Preferiti

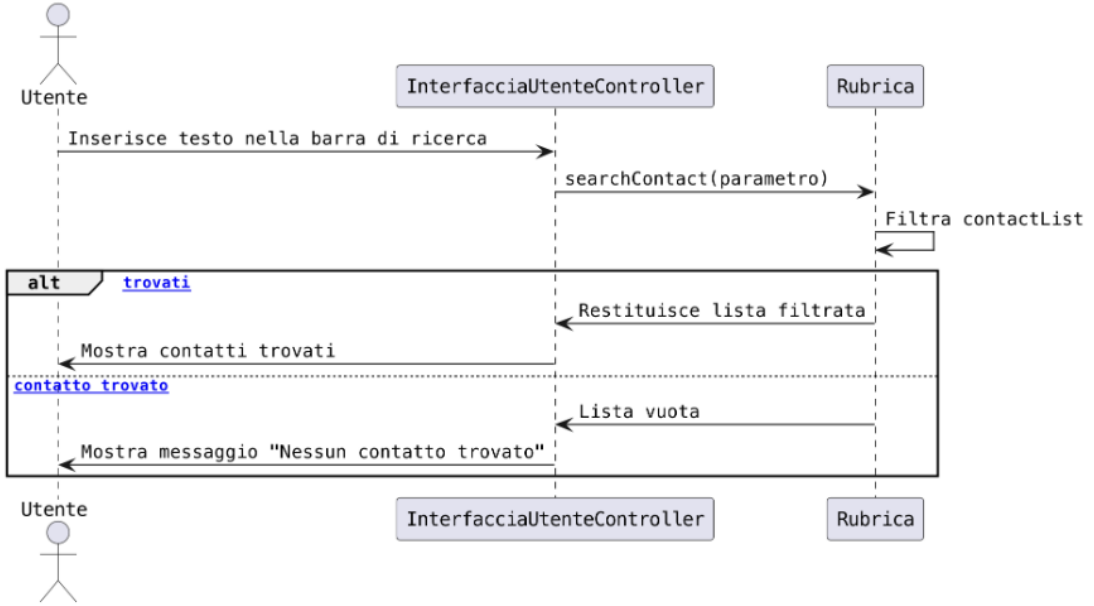


Figura 6: Ricerca Contatto

DIAGRAMMA DELLE CLASSI

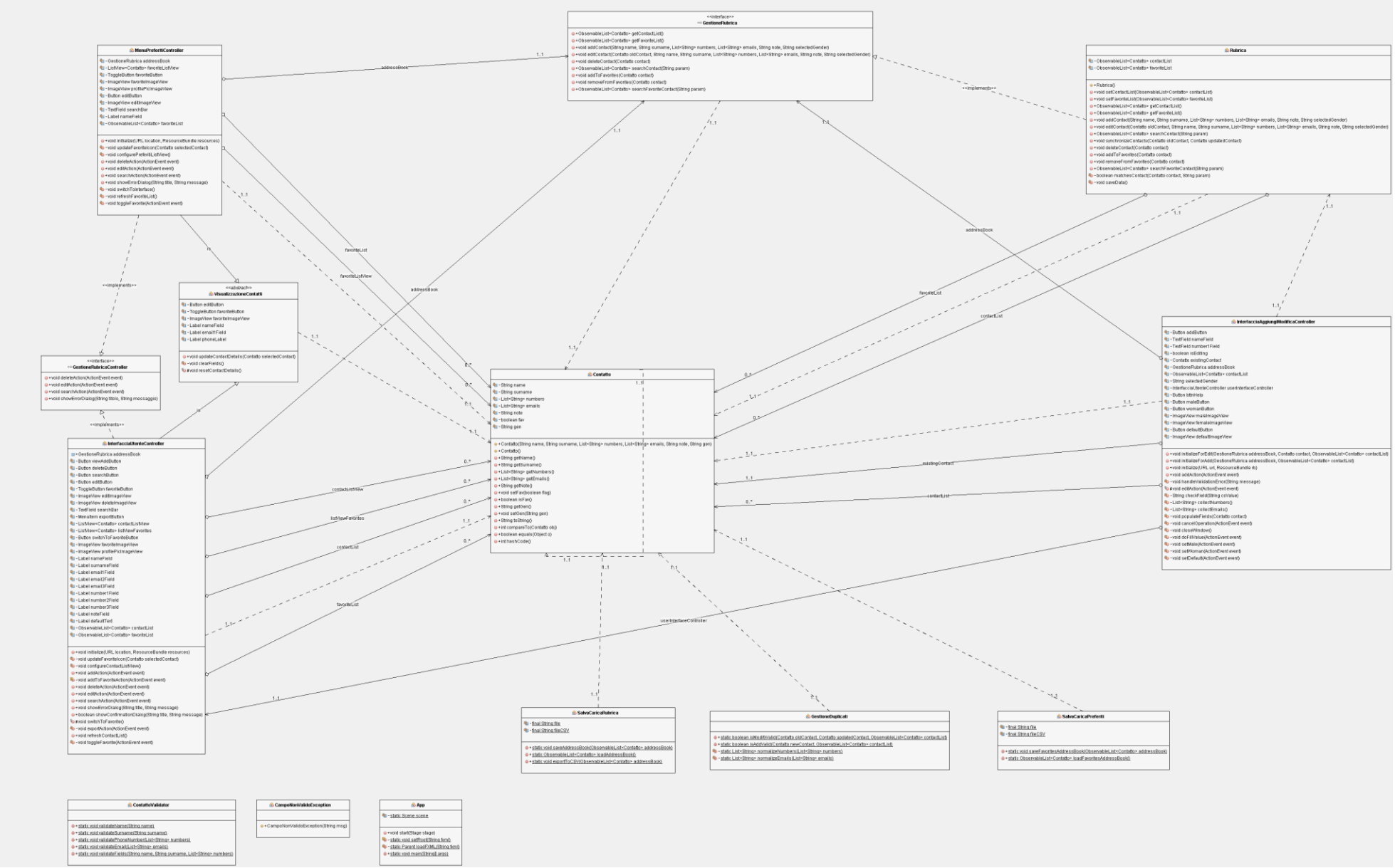


DIAGRAMMA DELLE ATTIVITÀ

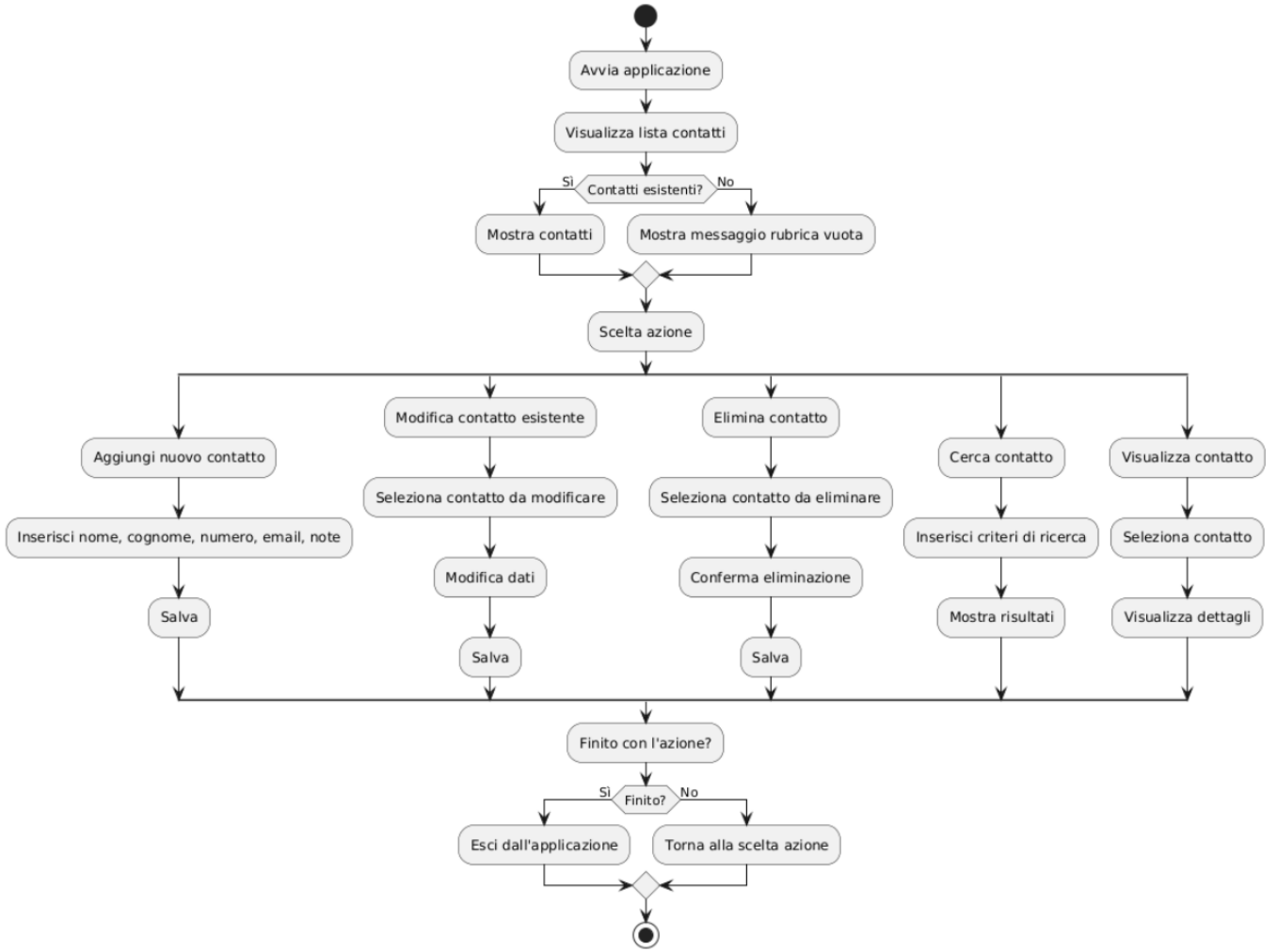


Figura 8: Diagramma delle Attività

CASI DI TEST

- Esempi di **casi di test unitari** (unit test cases, realizzato per verificare che le singole unità di codice, come funzioni o metodi, funzionino correttamente in isolamento.

Test	Rubrica
TestItems	testEditContact_ValidEdit_ShouldUpdateContact

Input	rubrica = new Rubrica();
	rubrica.setContactList(FXCollections.observableArrayList());
	rubrica.setFavoriteList(FXCollections.observableArrayList());
	Contatto oldContact = new Contatto("Dati");
	rubrica.getContactList().add(oldContact);
	String newName = "";
	String newSurname = "";
	List<String> newNumbers = Arrays.asList("");
	List<String> newEmails = Arrays.asList("");
	String newNote = "";
	String newGen = "";
	rubrica.editContact(oldContact, newName, newSurname, newNumbers, newEmails, newNote, "newGen");

Oracle	contactList.size() == 1;
	updatedContact.getName() == newName;
	updatedContact.getSurname() == newSurname;
	updatedContact.getNumbers() == newNumbers;
	updatedContact.getEmails() == newEmails;
	updatedContact.getNote() == newNote;
	updatedContact.getNote() == newGen;

Test	Rubrica
TestItems	AddContact_ValidContact_ShouldAddSuccessfully

Input	rubrica = new Rubrica();
	rubrica.setContactList(FXCollections.observableArrayList());
	rubrica.setFavoriteList(FXCollections.observableArrayList());
	rubrica.addContact(name, surname, numbers, emails, note, gen);

Oracle	contactList.size() == 1;
	addedContact.getName() == name;
	addedContact.getSurname() == surname;
	addedContact.getNumbers() == numbers;
	addedContact.getEmails() == emails;
	addedContact.getNote() == note;

ANALISI DI COESIONE DEI MODULI DEL SISTEMA

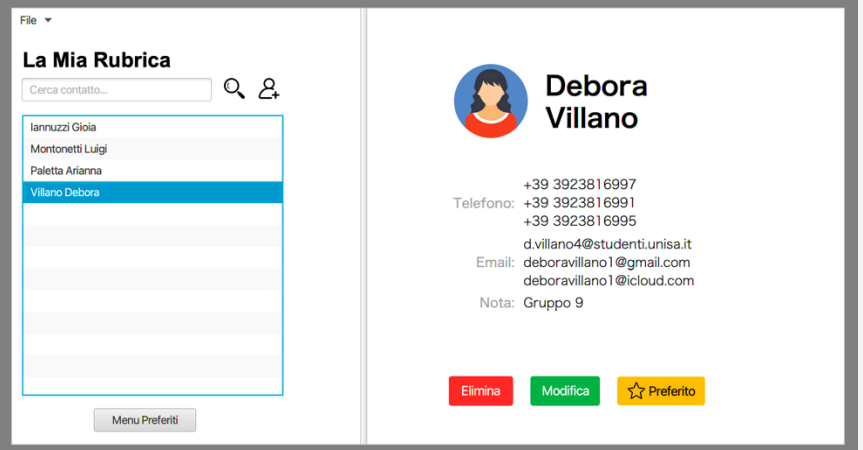
Modulo	Livello di Coesione	Descrizione
`Contatto`	Funzionale	Rappresenta l'entità "Contatto" con tutte le sue proprietà (nome, cognome, numeri, email, note, genere).
`Rubrica`	Funzionale	Implementa tutte le operazioni relative alla gestione della rubrica (aggiunta, modifica, eliminazione, ricerca, gestione dei preferiti).
`InterfacciaUtenteController`	Funzionale	Gestisce tutte le interazioni con l'utente attraverso l'interfaccia grafica.
`MenuPreferitiController`	Funzionale	Le funzionalità si concentrano sulla gestione dei contatti preferiti.
`InterfacciaAggiungiModificaController`	Procedurale	Gestisce il flusso di lavoro per aggiungere o modificare un contatto. Le operazioni includono la raccolta dei dati e il passaggio del contatto aggiornato alla rubrica.
`SalvaCaricaRubrica`	Funzionale	Lo scopo principale è gestire il salvataggio e il caricamento di dati (in formato JSON o CSV).
`SalvaCaricaPreferiti`	Funzionale	Simile a SalvaCaricaRubrica, ma specifico per i contatti preferiti.
`GestioneDuplicati`	Logica	Le operazioni effettuate sono simili, ma in contesti diversi.
`CampoNonValidoException`	Funzionale	Segnala errori relativi alla validazione dei campi, con metodi dedicati a costruire e restituire messaggi di errore.
`ContattoValidator`	Logica	Controlla la validità dei campi (nome, cognome, numeri di telefono, email) per garantire l'integrità dei dati.

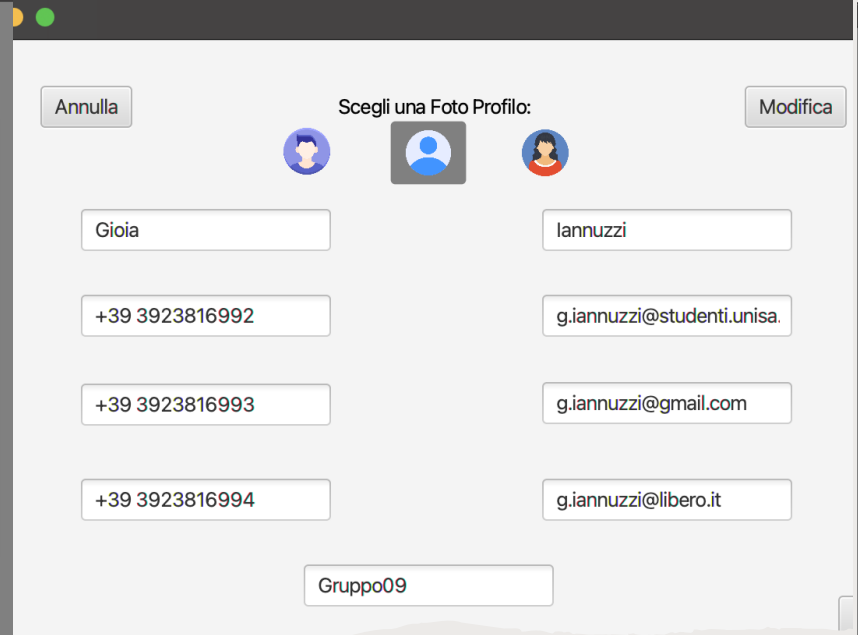
Tabella 15: Analisi della Coesione dei Moduli del Sistema

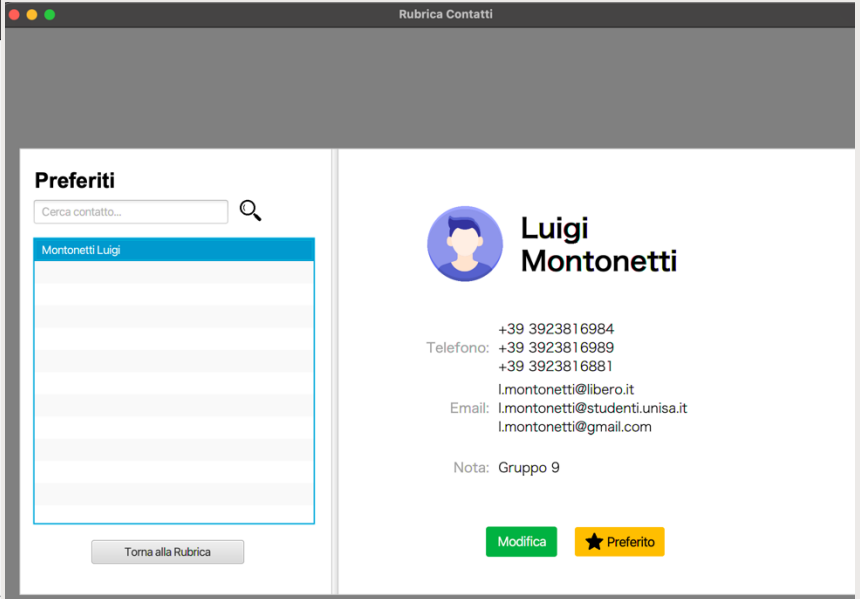
ANALISI DELL'ACCOPPIAMENTO DEI MODULI DEL SISTEMA

Modulo	Livello di Accoppiamento	Descrizione
`Contatto`	Per Dati	Si è scelto un accoppiamento minimo per trasferire solo i dati necessari, garantendo che la classe resti focalizzata sulla rappresentazione del modello.
`Rubrica`	Per Timbro, Per Dati	Le sue dipendenze sono coerenti con il suo ruolo come gestore centrale dei dati. Tuttavia, alcuni metodi come `searchContact` dipendono da parametri esterni.
`InterfacciaUtenteController`	Per Timbro, Per Dati	L'accoppiamento con l'interfaccia GestioneRubrica è necessario per trasferire dati complessi, come le liste osservabili (rubriche) o dei numeri, tra l'interfaccia grafica e il modello, garantendo sincronizzazione dinamica e aggiornamenti in tempo reale.
`MenuPreferitiController`	Per Timbro, Per Dati	Le liste osservabili sono passate per facilitare la gestione dinamica dei dati preferiti, minimizzando la complessità di sincronizzazione con la rubrica principale.
`InterfacciaAggiungiModificaController`	Per Dati	Si è scelto un accoppiamento per dati essenziali per garantire una gestione diretta delle operazioni di inserimento e modifica.
`SalvaCaricaRubrica`	Per Timbro	L'accoppiamento per timbro è necessario per scambiare in modo sicuro e consistente liste osservabili tra la rubrica dei preferiti e i file JSON, sfruttando la libreria Jackson.
`SalvaCaricaPreferiti`	Per Timbro	L'accoppiamento è il medesimo di SalvaCaricaRubrica.
`GestioneDuplicati`	Per Timbro, Per Dati	L'accoppiamento consente di verificare duplicati direttamente sulle liste osservabili.
`CampoNonValidoException`	Per Dati	Un accoppiamento per dati è sufficiente per segnalare errori specifici.
`ContattoValidator`	Per Dati	L'accoppiamento consente un controllo rigoroso della validità dei dati, segnalando errori attraverso eccezioni, mantenendo la separazione dei compiti.

Tabella 16: Analisi dell'Accoppiamento dei Moduli del Sistema







FINE

Gruppo 9

- *0612707416 - Gioia Iannuzzi*
- *0612708273 - Luigi Montonetti*
- *0612707862 - Arianna Paletta*
- *0612708294 - Debora Villano*